

CLAIMS

What is claimed is:

- 1
2
3
4
5
6
1. An apparatus comprising:
a peripheral bus coupled to a peripheral unit to transfer peripheral
information including a command message specifying a peripheral operation; and
a processing slice coupled to the peripheral bus to execute a plurality of
threads, the plurality of threads including a first thread sending the command
message to the peripheral unit.
- 1
2
2. The apparatus of claim 1 wherein the peripheral unit is one of an
input device and an output device.
- 1
2
3. The apparatus of claim 1 wherein the peripheral operation is one of
an input operation and an output operation.
- 1
2
3
4. The apparatus of claim 1 wherein the command messages includes
at least one of a message content, a peripheral address identifying the peripheral
unit, and a command code specifying the peripheral operation.
- 1
2
3
5. The apparatus of claim 1 wherein the peripheral information
includes a response message sent from the peripheral unit to the processing slice,
the response message indicating the peripheral operation is completed.

1 6. The apparatus of claim 5 wherein the response message includes at
2 least one of a thread identifier identifying the first thread, an operation result of
3 the peripheral operation, a data register address specifying a data register in the
4 processing slice to store the operation result, and a length indicator indicating
5 length of the response message.

1 7. The apparatus of claim 6 wherein the peripheral bus comprises:
2 a bi-directional bus to transfer the command message from the processing
3 slice to the peripheral unit and the response message from peripheral unit to the
4 processing slice.

1 8. The apparatus of claim 1 wherein the processing slice disables the
2 first thread after sending the command message if the command message is a wait
3 instruction.

1 9. The apparatus of claim 1 wherein the first thread continues to
2 execute after sending the command message if the command message is a non-
3 wait instruction.

1 10. The apparatus of claim 8 wherein the processing slice enables the
2 first thread after receiving the response message from the peripheral unit if the
3 first thread was disabled.

1 11. The apparatus of claim 1 wherein the processing slice comprises:

2 an instruction processing unit to process instructions fetched from a
3 program memory; and
4 a thread control unit coupled to the instruction processing unit to manage
5 initiating and termination of at least one of the plurality of threads.

1 12. The apparatus of claim 11 wherein the processing slice further
2 comprises:

3 a memory access unit coupled to the instruction processing unit to provide
4 access to one of a plurality of data memories via a data memory switch, the
5 memory access unit having a plurality of data base registers, each of the data base
6 registers corresponding to each of the threads; and

7 a functional unit coupled to the instruction processing unit to perform an
8 operation specified in one of the instructions; and

9 a register file coupled to the instruction processing unit and the peripheral
10 unit having a plurality of data registers, each of the data registers corresponding to
11 each of the threads.

1 13. The apparatus of claim 12 wherein the instruction processing unit
2 comprises:

3 an instruction fetch unit to fetch the instructions from the program memory
4 using a plurality of program counters, each program counter corresponding to each
5 of the threads;

6 an instruction buffer coupled to the instruction fetch unit to hold the
7 fetched instructions; and

8 an instruction decoder and dispatcher coupled to the instruction buffer to
9 decode the instructions and dispatch the decoded instructions to one of the
10 memory access unit, the functional unit, and the peripheral unit;

11 wherein the instructions are executed concurrently in a clock cycle.

1 14. A method comprising:

2 transferring peripheral information to a peripheral unit via a peripheral bus,
3 the peripheral information including a command message specifying a peripheral
4 operation; and

5 executing a plurality of threads by a processing slice, the plurality of
6 threads including a first thread sending the command message to the peripheral
7 unit.

1 15. The method of claim 14 wherein the peripheral unit is one of an
2 input device and an output device.

1 16. The method of claim 14 wherein the peripheral operation is one of
2 an input operation and an output operation.

1 17. The method of claim 14 wherein the command messages includes
2 at least one of a message content, a peripheral address identifying the peripheral
3 unit, and a command code specifying the peripheral operation.

1 18. The method of claim 14 wherein the peripheral information
2 includes a response message sent from the peripheral unit to the processing slice,
3 the response message indicating the peripheral operation is completed.

1 19. The method of claim 18 wherein the response message includes at
2 least one of a thread identifier identifying the first thread, an operation result of
3 the peripheral operation, a data register address specifying a data register in the
4 processing slice to store the operation result, and a length indicator indicating
5 length of the response message.

1 20. The method of claim 19 wherein transferring the peripheral
2 information comprises:

3 transferring the command message from the processing slice to the
4 peripheral unit and the response message from peripheral unit to the processing
5 slice via a bi-directional bus.

1 21. The method of claim 14 wherein executing the plurality of threads
2 comprises disabling the first thread after sending the command message if the
3 command message is a wait instruction.

1 22. The method of claim 14 wherein executing the plurality of threads
2 comprises continuing executing the first thread after sending the command
3 message if the command message is a non-wait instruction.

1 23. The method of claim 21 wherein executing the plurality of threads
2 comprises enabling the first thread after receiving the response message from the
3 peripheral unit if the first thread was disabled.

1 24. The method of claim 14 wherein executing the plurality of threads
2 comprises:

3 processing instructions fetched from a program memory by an instruction
4 processing unit;

5 managing initiating and termination of at least one of the plurality of
6 threads b a thread control unit.

1 25. The method of claim 24 wherein executing the plurality of threads
2 further comprises:

3 accessing to one of a plurality of data memories by a memory access unit
4 via a data memory switch, the memory access unit having a plurality of data base
5 registers, each of the data base registers corresponding to each of the threads;

6 performing an operation specified in one of the instructions by a functional
7 unit; and

8 storing data in a register file having a plurality of data registers, each of the
9 data registers corresponding to each of the threads.

1 26. The method of claim 24 wherein processing instructions
2 comprises:

3 fetching the instructions from the program memory using a plurality of
4 program counters by an instruction fetch unit, each program counter
5 corresponding to each of the threads;

6 holding the fetched instructions in an instruction buffer;

7 decoding the instructions and dispatching the decoded instructions by an
8 instruction decoder and dispatcher to one of the memory access unit, the
9 functional unit, and the peripheral unit; and
10 executing the instructions concurrently in a clock cycle.

1 27. A processing system comprising:
2 a plurality of banks of data memory;
3 a data memory switch coupled to the banks to data memory;
4 a program memory to store a program;
5 a peripheral bus coupled to a peripheral unit to transfer peripheral
6 information including a command message specifying a peripheral operation; and
7 a processing slice coupled to the peripheral bus to execute a plurality of
8 threads, the plurality of threads including a first thread sending the command
9 message to the peripheral unit.

1 28. The processing system of claim 27 wherein the peripheral unit is
2 one of an input device and an output device.

1 29. The processing system of claim 27 wherein the peripheral
2 operation is one of an input operation and an output operation.

1 30. The processing system of claim 27 wherein the command messages
2 includes at least one of a message content, a peripheral address identifying the
3 peripheral unit, and a command code specifying the peripheral operation.

1 31. The processing system of claim 27 wherein the peripheral
2 information includes a response message sent from the peripheral unit to the
3 processing slice, the response message indicating the peripheral operation is
4 completed.

1 32. The processing system of claim 31 wherein the response message
2 includes at least one of a thread identifier identifying the first thread, an operation
3 result of the peripheral operation, a data register address specifying a data register
4 in the processing slice to store the operation result, and a length indicator
5 indicating length of the response message.

1 33. The processing system of claim 32 wherein the peripheral bus
2 comprises:

3 a bi-directional bus to transfer the command message from the processing
4 slice to the peripheral unit and the response message from peripheral unit to the
5 processing slice.

1 34. The processing system of claim 27 wherein the processing slice
2 disables the first thread after sending the command message if the command
3 message is a wait instruction.

1 35. The processing system of claim 27 wherein the first thread
2 continues to execute after sending the command message if the command message
3 is a non-wait instruction.

1 36. The processing system of claim 34 wherein the processing slice
2 enables the first thread after receiving the response message from the peripheral
3 unit if the first thread was disabled.

1 37. The processing system of claim 27 wherein the processing slice
2 comprises:

3 an instruction processing unit to process instructions fetched from a
4 program memory; and

5 a thread control unit coupled to the instruction processing unit to manage
6 initiating and termination of at least one of the plurality of threads.

1 38. The processing system of claim 37 wherein the processing slice
2 further comprises:

3 a memory access unit coupled to the instruction processing unit to provide
4 access to one of a plurality of data memories via a data memory switch, the
5 memory access unit having a plurality of data base registers, each of the data base
6 registers corresponding to each of the threads;

7 a functional unit coupled to the instruction processing unit to perform an
8 operation specified in one of the instructions; and

9 a register file coupled to the instruction processing unit and the peripheral
10 unit having a plurality of data registers, each of the data registers corresponding to
11 each of the threads.

1 39. The processing system of claim 38 wherein the instruction
2 processing unit comprises:

3 an instruction fetch unit to fetch the instructions from the program memory
4 using a plurality of program counters, each program counter corresponding to each
5 of the threads;

6 an instruction buffer coupled to the instruction fetch unit to hold the
7 fetched instructions; and

8 an instruction decoder and dispatcher coupled to the instruction buffer to
9 decode the instructions and dispatch the decoded instructions to one of the
10 memory access unit, the functional unit, and the peripheral unit;
11 wherein the instructions are executed concurrently in a clock cycle.

1 40. A processing system comprising:
2 a plurality of multi-thread processors;
3 a plurality of peripheral units;
4 a peripheral bus coupled to the peripheral units to transfer peripheral
5 information between the multi-thread processors and the peripheral units, the
6 peripheral information including a command message sent from one of the multi-
7 thread processors to one of the peripheral units by a thread executing a message
8 instruction.

1 41. A processing system comprising:

2 a multi-thread processor having program base registers and data base
3 registers;
4 at least one peripheral units;
5 a peripheral bus coupled to the at least one peripheral unit to transfer
6 peripheral information between the multi-thread processor and the at least one
7 peripheral unit, the peripheral information including a command message sent
8 from one of the multi-thread processors to one of the peripheral units by a thread
9 executing a message instruction.